(10) Question 1.

(2) Part a) What does volatile mean in context of computer memory ………………………

Volatile means if power is removed and then restored, the data contents will be lost (RAM).
Nonvolatile means if power is removed and then restored, the data contents will be lost (ROM).

(2) Part b) If R1 is 0xFFFFFFFF. Does the branch to **Happy** occur? ………

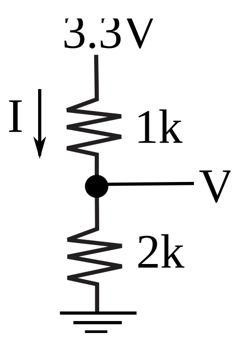
```
CMP R1,#3
BHS Happy
```

Yes, BHS will branch
4 billion is larger than 3

(2) Part c) Considering R0 as input and R1 as output, what is the mathematical relationship between R1 and R0? ………………………

```
LSLS R1,R0,#2
ADDS R1,R1,R0
```

R1 = 4*R0+R0 = 5*R0



(2) Part d) What is V? …………

V = 3.3V*2/3 = 2.2V

(2) Part e) What is I? ……………

I = 3.3V/3k = 1.1mA

(15) Question 2. There are two 100-element 8-bit signed global arrays, **X Y**.
Write a Cortex M assembly subroutine implementation of **Fill**. Follow AAPCS.

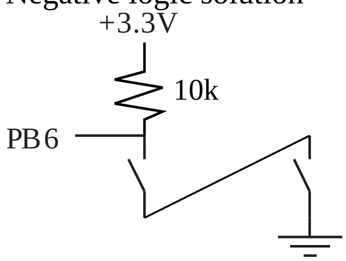
```
Fill:  MOVS R2,#0
Loop2: STRB R1,[R0,R2]
       ADDS R2,#1
       CMP  R2,#100
       BLT  Loop2  // could be BLO
       BX   LR
```

(20) Question 3. Write a C function to implement factorial. Return 0xFFFFFFFF if the calculation would overflow 32-bit unsigned math. Note that 12! is about 479 million, while 13! is about 6 billion. The function prototype is fixed and cannot be changed. **n**! is defined as 1*2*3*…***n**.

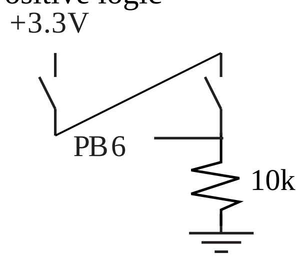
```
uint32_t Fact(uint32_t n){
  if(n > 12) return 0xFFFFFFFF;
  uint32_t result = 1;
  while(n){
    result = result*n;
    n--;
  }
  return result;
}
uint32_t Fact(uint32_t n){
  if(n <= 1) return 1; // end case
  if(n > 12) return 0xFFFFFFFF;
  return n*Fact(n-1);
}
```

(10) Question 4. Interface two **switches** to Port B. Put switches in series to achieve the "both" functionality. Use a resistor to create the passive value when switch not pressed. Use the switch to create the active value when switch is pressed.

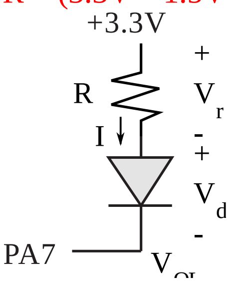
Negative logic solution



Positive logic



(10) Question 5. Interface an **LED** to Port A pin 7 using negative logic. KCL: the current through resistor equals the current through LED. KVL: 3.3V = V_r+V_d+V_{OL}. Ohms: V_r = I*R. Solve for R = (3.3V – V_d – V_{OL})/I
R = (3.3V – 1.5V – 0.3V)/3ma = 1500mV/3mA = 500ohms



(10) Question 6. Show the contents of all five registers after we execute this code.

```
  MOVS R0,#0
  MOVS R1,#1
  MOVS R2,#2
  MOVS R3,#3
  MOVS R4,#4
  PUSH {R2}
  PUSH {R1,R3,R4}
  LSLS R0,R3,R1
  POP  {R4}
  POP  {R1,R2,R3}
```

R0 = 6

R1 = 3

R2 = 4

R3 = 2

R4 = 1

```
    PUSH {R2}
//   SP->2
    PUSH { R1,R3,R4}
//   SP->1
//       3
//       4
//       2
     LSRS R0,R3,R4 // R0 = 3<<1 = 3*2 = 6
     POP {R4}  // R4 = 1
//   SP->3
//       4
//       2
//   POP  {R1,R2,R3} //R1=3, R2=4, R3=2
```

(5) Question 7. Assume there is an array pointed to by R0. *Hint: look carefully at the memory addresses in the following figure.*

Address	Contents	
0x20201000	**0x90**	**<- R0**
0x20201001	**0x92**	
0x20201002	**0x92**	**<- R0+2**
0x20201003	**0x93**	
0x20201004	**0x94**	
0x20201005	**0x95**	

Assume register R0 equals 0x20201000. What is the value of R2 after executing the following instructions?

```
MOVS  R1,#2
LDRSH R2,[R0,R1]
```

R2= 0xFFFF9392 (sign extend)

```
LDRSH R2,[R0,R1]
```

(20) Question 8. PB7 is a positive logic switch input, and PB2 is a positive logic LED output.

```
main: MOVS R0,#0
      BL   Clock_Init80MHz // 12.5ns bus cycle
      BL   Init // given, do not write
      LDR  R5,=GPIOB_DIN31_0
      LDR  R6,=GPIOB_DOUTSET31_0
      LDR  R7,=GPIOB_DOUTCLR31_0
      MOVS R4,#4     // bit2 mask
loop: MOVS R2,#0x80 // bit 7 mask, Delayms may destroy R2
      LDR  R1,[R5]  // read all
      ANDS R1,R1,R2 // check bit 7
      BEQ  low
high: STR  R4,[R6]  // LED on
      BL   Delayms  // 2ms
      BL   Delayms
      STR  R4,[R7]  // LED off
      BL   Delayms  // 1ms
      B    loop
low:  STR  R4,[R6]  // LED on
      BL   Delayms  // 1ms
      STR  R4,[R7]  // LED off
      BL   Delayms  // 2ms
      BL   Delayms
      B    loop
```

put.
main
ust